



Università degli Studi di Bologna

Corso di Laurea in Ingegneria Informatica

Introduzione

Ingegneria del Software T

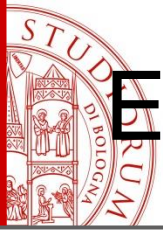
Prof. MARCO PATELLA

Dipartimento di Informatica – Scienza e Ingegneria (DISI)



Argomenti

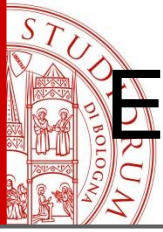
- Evoluzione dello sviluppo del software
- Prodotto software
- Processo di sviluppo del software
 - Attività
 - Modelli
 - Prototipi
 - Linguaggi di modellazione – UML
- Fattori di qualità del software



Evoluzione dello Sviluppo del Software

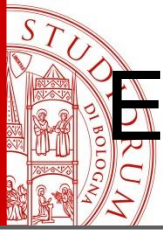
- Seconda metà degli anni '60
l'avvento dei calcolatori di terza generazione (circuiti integrati), rende possibili applicazioni considerate in precedenza non realizzabili
- Si rende necessario
 - Progettare
 - Realizzare
 - Effettuare la manutenzione (*manutenere*)

sistemi software di grandi dimensioni



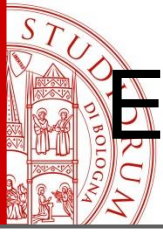
Evoluzione dello Sviluppo del Software

- Nel settore dell'hardware, netto, progressivo:
 - aumento delle prestazioni
 - riduzione dei costi



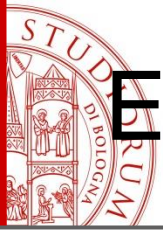
Evoluzione dello Sviluppo del Software

- Lo sviluppo del software deve evolvere in una **disciplina ingegneristica**, con basi
 - teoriche
 - metodologiche
- Nel 1968, durante una conferenza sull'evoluzione del software, viene coniato il termine “***ingegneria del software***”



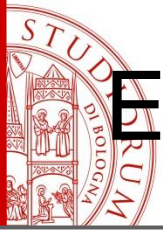
Evoluzione dello Sviluppo del Software

- Nel settore del software:
 - Pochissimi i progetti terminati nei tempi stabiliti
 - Rari i progetti che non sfondano il budget
 - Sistemi generalmente pieni di difetti
 - Sistemi così rigidamente strutturati che è praticamente impossibile apportare significativi cambiamenti senza doverli riprogettare totalmente
- “***Crisi del software***”



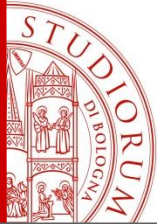
Evoluzione dello Sviluppo del Software

Anni	Base tecnologica	Problema	Rivoluzione software
1950	primi calcolatori in serie	produttività	Assembler
1960	calcolatori mono-utente in batch	produttività	linguaggi di alto livello (Fortran, Cobol, Algol)
1970	tecnologia dei compilatori	complessità	programmazione strutturata (Pascal, C)
1980	calcolatori sempre più potenti	complessità	programmazione modulare ADT (Ada, Modula-2)
1990	workstation, reti, interfacce grafiche,...	enorme aumento della complessità	programmazione orientata agli oggetti (Smalltalk, C++, Eiffel, CLOS, Object Pascal, Java, C#, ...)



Evoluzione dello Sviluppo del Software

- Da **lavoro individuale** e creativo (software come arte)
- A lavoro di **piccoli gruppi** specializzati (livello artigianale)
- Sino a lavoro di **gruppi** anche **di grosse dimensioni** in cui è necessaria un'opera di pianificazione e coordinamento (livello industriale)



Software

- Il software è *astratto* e *intangibile*
- Esso non è vincolato da materiali o governato da leggi fisiche
- Da un lato questo semplifica il lavoro dei progettisti che non hanno limitazioni fisiche
- Dall'altro la naturale mancanza di vincoli porta il software a diventare estremamente complicato e molto difficile da capire

Software

- Spesso è facile perdere la “connessione” con il mondo reale che il software deve gestire e i malfunzionamenti vengono sottovalutati
- Un software *mal progettato* e *mal sviluppato* può funzionare in modo non previsto causando parecchi danni e conseguenze a volte anche catastrofiche
- Zdnet (2007) riporta una classifica dei dieci casi più gravi (senza perdite di vite umane)



Software difettoso (10)

- **2007 - aeroporto di Los Angeles in tilt**
 - Per colpa di un software, 17mila voli rimangono a terra allo scalo aereo di Los Angeles
 - Tutto per una scheda di rete che invece di spegnersi continua ad immettere dati sbagliati all'interno del sistema
 - Il sistema va in tilt e nessuno può entrare o uscire dallo scalo per otto ore



Software difettoso (9)

- **1999 - Siemens ed i passaporti britannici**
 - Nell'estate di quell'anno mezzo milione di cittadini britannici non ricevono il loro passaporto in tempo per colpa di un sistema computerizzato Siemens non sufficientemente testato
 - Centinaia di persone non poterono andare in vacanza e il governo è stato costretto a pagare milioni di sterline in risarcimenti



Software difettoso (8)

- **2006 - dati scomparsi in Gran Bretagna**
 - Per un “disguido tecnico” è scomparsa dal database del fisco di Sua Maestà tutta una serie di informazioni sensibili sparite con tanto di indirizzi, estremi bancari, nomi, cognomi e date di nascita dei diretti interessati
 - La perdita di dati riservati con gravissime conseguenze pratiche ha toccato 25 milioni di persone



Software difettoso (7)

- **1999-2000 - il baco del millennio**
 - Quando scatta la mezzanotte, il 31 dicembre 1999 l'annunciato disastro in realtà non arriva
 - Ma il Millennium Bug si piazza in classifica per la quantità di denaro spesa per anticipare il danno: oltre 500 miliardi di euro



Software difettoso (6)

- **2004 - EDS e (CSA) l'agenzia per il sostegno ai bambini**
 - Per colpa dell'introduzione di un nuovo sistema informatico durante la ristrutturazione della CSA britannica quasi due milioni di persone vengono pagate in eccesso dal gigante di servizi commerciali EDS, mentre 700mila ricevono pagamenti inferiori a quelli dovuti
 - Risultato: il collasso dell'agenzia e un costo per i contribuenti che superò il miliardo di sterline



Software difettoso (5)

- **1998 - osservazione del clima su Marte e i problemi di misurazioni**
 - Due navicelle spaziali, il Mars Climate Orbiter e il Mars Polar Lander fanno parte di un programma per studiare il clima e il tempo atmosferico sul pianeta rosso
 - Per un problema di navigazione il Lander volava troppo basso e venne distrutto: tutta colpa di due diversi sistemi di misurazione dell'altezza usati sulla sonda (sistema anglosassone vs. metrico decimale)



Software difettoso (4)

- **2006 - software incompatibili sull'Airbus A380**
 - Errore di comunicazione fra due delle organizzazioni del consorzio per la produzione di Airbus
 - Una fabbrica usa una versione vecchia del software CATIA mentre un'altra usa l'ultima versione
 - Quando si mettono insieme le due metà dell'aereo, i software non coincidono e le connessioni interne risultano impossibili
 - Perfino i cavi sono incompatibili e vanno tutti sostituiti
 - La “disattenzione” costò moltissimo (nessuno ha mai voluto quantificare...) e ritardò il programma di un anno



Software difettoso (3)

- **1996 - esplosione dell'Ariane 5**
 - Il nuovissimo razzo con annesso satellite europeo salta in aria pochi secondi dopo il lancio per il suo viaggio inaugurale
 - L'autodistruzione viene provocata dal software interno che, per un errore, cerca di inserire un numero float da 64 bit nello spazio di uno intero da 16
 - Costo del disastro: 8 miliardi di dollari



Software difettoso (2)

- **1990 – Il collasso della rete AT&T**
 - Il 15/1, 75 milioni di chiamate telefoniche perse dopo un piccolo malfunzionamento meccanico di un singolo switch in uno dei 144 centralini di smistamento di AT&T negli USA
 - Al riavvio, il centralino ha inviato un messaggio agli altri centralini, provocandone il riavvio
 - In origine si pensò a un attacco di hacker, ma il responsabile fu un errore in una singola linea di codice, inserita durante un cospicuo upgrade software
 - American Airlines ha stimato la perdita di circa 200K prenotazioni aeree



Software difettoso (1)

- **1983 – Sfiolata la Terza Guerra Mondiale**
 - Una notte di fine settembre il sistema satellitare di preallarme russo individuò il lancio a distanza ravvicinata di cinque missili intercontinentali da parte degli Stati Uniti
 - Grazie alla decisione presa da Stanislav Petrov (che era al comando quella notte) di liquidare quei lanci come semplici errori del sistema non è scoppiata una guerra nucleare
 - I danni sono stati pochissimi rispetto al rischio di una guerra



Prodotto Software

- **Codice** che, quando eseguito, svolge una funzione prestabilita con prestazioni prestabilite
- **Strutture dati** mediante le quali il codice tratta adeguatamente le informazioni
- **Documenti** che descrivono le operazioni e l'uso del prodotto software:
 - documentazione tecnica
 - manualistica di installazione
 - manualistica di utilizzo
 - ...



Prodotto Software

- Può essere realizzato
 - per un particolare cliente
 - per il mercato in generale
- Può fare parte di un **sistema di elaborazione** che comprende
 - sia la parte software
 - sia la parte hardware



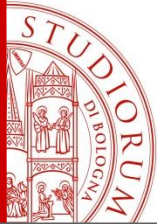
Prodotto Software

- Si sviluppa, non si fabbrica
 - Cliente
 - Sviluppatore/i
 - Processo di sviluppo
 - Linguaggio e strumenti di modellazione
- Non “si consuma”...
- Struttura
- Caratteristiche



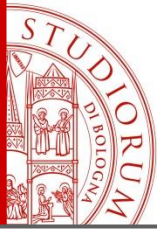
Sviluppo del Software

- Esistono molti esempi di
 - Progetti falliti (alcuni già visti)
 - Tempi e costi non rispettati
 - Soluzioni non corrette
 - Sistemi non gestibili
- Il successo di un progetto
 - Non può essere garantito
 - È determinato
 - Principalmente da fattori umani
 - Solo secondariamente da scelte tecnologiche



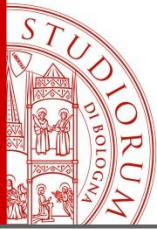
Principali Cause di Fallimento

- Necessità del **cliente** comprese male o in modo insufficiente
- Requisiti del **cliente** troppo instabili
- Mancanza di cooperazione tra **cliente** e **sviluppatori**
- Attese non realistiche da parte del **cliente**
- Mancanza di benefici per il **cliente**
- Fornitura insufficiente di risorse da parte del **cliente**



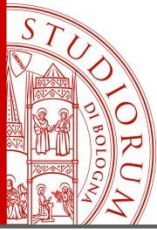
Principali Cause di Fallimento

- **Sviluppatori** non all'altezza delle attività in cui sono coinvolti
- Capacità e conoscenza degli **sviluppatori** sono i fattori che maggiormente contribuiscono
 - alla qualità del software
 - alla produttività
- Buoni **sviluppatori** possono fornire una soluzione, ma ottimi sviluppatori possono fornire
 - soluzioni migliori
 - più velocemente
 - a minor costo



Ingegneria del Software

- Una disciplina
 - tecnologica
 - gestionale
- che permette di affrontare in modo
- sistematico
 - quantificabile (in termini di tempi e costi)
- lo sviluppo e la manutenzione
dei prodotti software



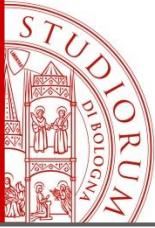
Ingegneria del Software

Software Engineering is an engineering discipline concerned with theories, methods and tools for professional software development

Software Engineering is concerned with **all aspects** of software production from the early stage of system specification to the system maintenance/incremental development after it has gone into use

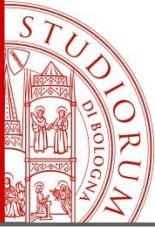


Da: Software Engineering 8 – I.Sommerville - Addison Wesley



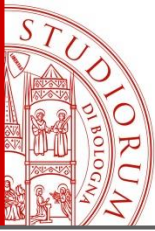
Ingegneria del Software

- L'ingegneria del software si occupa quindi:
 - del nuovo sistema che deve essere costruito (prodotto)
 - del processo di sviluppo
- Si ha la necessità di ingegnerizzare entrambi questi elementi:
 - un prodotto ingegnerizzato male è poco usabile, estendibile e manutenibile
 - allo stesso modo un processo di sviluppo sbagliato o mal concepito porta alla creazione di un prodotto di scarsa qualità
- Tutto questo porta alla necessità di creare *modelli* sia per il processo che per il prodotto



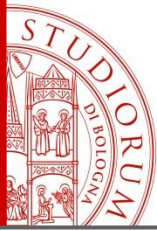
Ingegneria del Software

- Mettere in grado lo sviluppatore di affrontare la **complessità** dei grossi progetti
 - Sviluppare prodotti con le caratteristiche di qualità desiderate
 - Gestire le risorse (specie quelle umane) in modo produttivo
 - Rispettare i vincoli economici e di tempo specificati



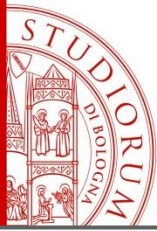
Ingegneria Tradizionale

- Il problema dello sviluppo del software è un problema di **gestione della complessità**
- In altri settori, l'uomo ha imparato bene a progettare e costruire sistemi complessi, come un edificio, un'automobile, un calcolatore, un impianto industriale
- Quali sono i principi usati nell'ingegneria tradizionale per affrontare la complessità?



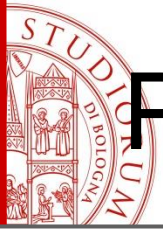
Principi Usati nell'Ingegneria Tradizionale per Affrontare la Complessità

- **Suddivisione del progetto in sottoprogetti** (moduli), e così via sino ad avere moduli facilmente progettabili
- **Utilizzo di parti e sottosistemi già pronti**
 - Sviluppati in casa
 - Comprati da fornitori esterni
- **Standardizzazione dei componenti**
in modo che possano essere facilmente collegabili e intercambiabili



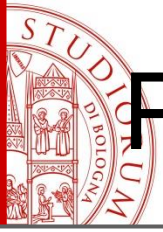
Principi Usati nell'Ingegneria Tradizionale per Affrontare la Complessità

- **Utilizzo del calcolatore** come ausilio per
 - La progettazione
 - L'esecuzione di compiti ripetitivi
 - I calcoli
- Lo sviluppo di un progetto non è lasciato al caso o all'estro di pochi progettisti, ma è un'**attività in larga misura sistematica**



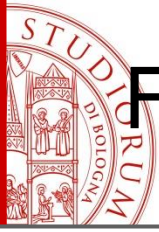
Processo di Sviluppo del Software

- **Insieme strutturato di attività** che regolano lo sviluppo di un prodotto software
- La struttura del processo di sviluppo può avere una forte influenza
 - sulla **qualità**
 - sul **costo**
 - sui **tempi di realizzazione**del prodotto



Processo di Sviluppo del Software

- Stabilisce un **ordine di esecuzione delle attività**
- Specifica quali **elaborati** devono essere forniti e quando
- Assegna le **attività** agli sviluppatori
- Fornisce criteri per
 - **monitorare** il progresso dello sviluppo
 - **misurarne** i risultati
 - **pianificare** i progetti futuri
- Copre l'intero **ciclo di vita del software**



Fasi del Processo di Sviluppo del Software

1. Studio di fattibilità
2. Analisi dei requisiti
3. Analisi del problema
4. Progettazione
5. Realizzazione e collaudo dei moduli
6. Integrazione e collaudo del sistema
7. Installazione e *training*
8. Utilizzo e manutenzione



Processo di Sviluppo del Software

Studio di Fattibilità

- Valutazione preliminare dei costi e dei benefici per stabilire se si debba avviare lo sviluppo dell'applicazione
 - Alternative possibili
 - Scelte più ragionevoli
 - Risorse finanziarie e umane necessarie per l'attuazione del progetto
- Il contenuto di questa fase risulta largamente variabile da caso a caso, in funzione del tipo di prodotto e dei ruoli del committente e del produttore dell'applicazione



Processo di Sviluppo del Software

Studio di Fattibilità

- Il risultato della fase di studio di fattibilità è un documento che dovrebbe contenere le seguenti informazioni:
 - definizione preliminare del problema
 - possibili scenari che illustrino eventuali diverse strategie di soluzione
 - costi, tempi e modalità di sviluppo per le diverse alternative
- Il committente può allocare risorse finanziarie perché venga condotto uno studio di fattibilità completo



Processo di Sviluppo del Software

Analisi dei Requisiti

- Un **requisito** è la descrizione
 - di un **comportamento** del sistema (**requisito funzionale**)
 - di un **vincolo** (**requisito non funzionale**)
 - sul comportamento del sistema
 - sullo sviluppo del sistema



Processo di Sviluppo del Software

Analisi dei Requisiti

- Applicazione per la gestione di uno sportello tipo bancomat:
 - **Requisiti funzionali:**
 - Il sistema dovrà validare la tessera inserita dal cliente
 - Il sistema dovrà validare il PIN inserito dal cliente
 - Il sistema dovrà erogare non più di 500 € alla stessa tessera nell'arco delle 24 ore
 - ...
 - **Requisiti non funzionali:**
 - Il sistema dovrà essere scritto in C++
 - Il sistema dovrà validare la tessera entro 3 secondi
 - Il sistema dovrà validare il PIN entro 3 secondi
 - Il sistema non dovrà consentire l'accesso a malintenzionati
 - ...



Processo di Sviluppo del Software

Analisi dei Requisiti

- Si stabiliscono
 - **funzionalità** (requisiti funzionali)
 - **vincoli** (requisiti non funzionali)
 - **obiettivi**

consultando gli utenti (analisi congiunta)

“Knowledge of what users really want often is the single most important factor in the failure or success of a software project.

It’s also one of the most neglected factors”

- Johnson, Skoglund, and Wisniewsky



Processo di Sviluppo del Software

Specifica dei Requisiti

1. Formalizzazione dei requisiti

- funzionalità, vincoli, interfaccia utente e qualsiasi altra caratteristica che il sistema dovrà possedere per soddisfare le necessità del cliente

2. Redazione di un documento di Specifica dei Requisiti Software (SRS – *Software Requirement Specification*)

- elenca tutti i requisiti che il sistema deve soddisfare
- deve essere completo, preciso, consistente, non ambiguo e comprensibile sia al cliente, sia allo sviluppatore

3. Convalida delle specifiche

- le specifiche devono essere riviste con il cliente per convalidarle



Processo di Sviluppo del Software

Specifica dei Requisiti

- SRS definisce
 - quali funzionalità deve fornire il sistema
 - **NON** come sono realizzate le funzionalità
- SRS è
 - un riferimento per l'attività di sviluppo del sistema
 - un contratto tra sviluppatori e cliente per la fornitura del sistema
 - deve essere approvato e sottoscritto dal committente
- **Stiamo realizzando il prodotto desiderato?**



Processo di Sviluppo del Software

Analisi del Problema

- L'analisi del problema parte sempre da una **attenta lettura** del testo che descrive i requisiti e degli scenari in cui essi sono articolati
- Un punto importante da ricordare è che l'analisi **non** si pone come obiettivo la descrizione di *quali proprietà strutturali* e *comportamentali* debba possedere il sistema risolvante: questo è l'obiettivo della fase di progetto



Processo di Sviluppo del Software

Analisi del Problema

- Dalla lettura degli scenari definiti nella fase di analisi dei requisiti si individua un primo insieme di problemi
- Si procede quindi ad effettuare l'*analisi dei problemi* avendo come obiettivo la definizione di un'**architettura logica** del sistema, basata sul **modello del dominio**



Processo di Sviluppo del Software

Analisi del Problema

- ***L'architettura logica*** del sistema deve esprimere “fatti” il più possibile “oggettivi” sul problema (***non sulla soluzione***) focalizzando l'attenzione su:
 - *Sottosistemi*
 - *Ruoli*
 - *Responsabilità*insiti negli scenari prospettati durante la descrizione dei requisiti



Processo di Sviluppo del Software

Analisi del Problema

- In sostanza *l'architettura logica* è un modello che descrive:
 - la struttura (insieme delle parti)
 - il comportamento atteso (dal tutto e da ciascuna parte)
 - le interazioni così come possono essere dedotte dai requisiti, dalle caratteristiche del problema e del dominio applicativo, *senza alcun riferimento* alle tecnologie realizzative



Processo di Sviluppo del Software

Progettazione

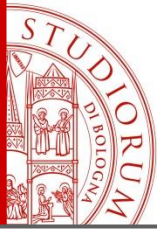
- Si definisce l'architettura generale (hardware e software) del sistema
- Si decompone l'architettura software in **uno o più programmi eseguibili**
- Per ogni programma, si descrivono:
 - le funzioni che svolge
 - le relazioni con gli altri programmi
- Si decompone ogni programma eseguibile in **più moduli**
- Per ogni modulo, si descrivono:
 - le funzioni che svolge
 - le relazioni con gli altri moduli



Processo di Sviluppo del Software

Progettazione

- Progettazione *object-based*
 - Tipi di dati astratti
 - Incapsulamento
- Progettazione *object-oriented*
 - Ereditarietà
 - Polimorfismo
- Progettazione generica rispetto ai tipi
 - *Template* in C++
 - Generici in Java e C#
- Gestione delle eccezioni
- Gestione degli eventi
- *Design pattern*



Processo di Sviluppo del Software

Progettazione

- Architettura del sistema
 - Modello *repository*
 - Modello *client-server*
 - Modello *abstract-machine*
- Controllo del sistema
 - Controllo centralizzato
 - Sistemi *event-driven*
- Concorrenza
- Progetto di interfacce utente
- Progetto di basi di dati



Processo di Sviluppo del Software

Progettazione

- Risultato: *documento di specifiche di progetto* nel quale la definizione dell'architettura software può anche essere data in maniera rigorosa, o addirittura formale, usando opportuni linguaggi di specifica di progetto
- Stiamo realizzando correttamente il prodotto?

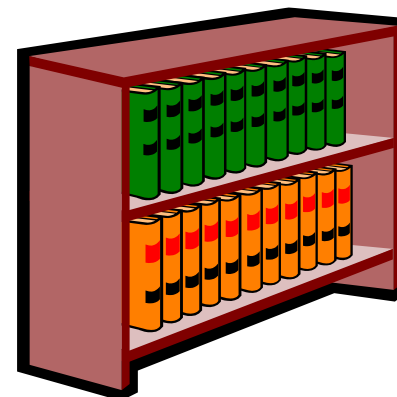


Realizzazione e Collaudo dei Moduli

- Il progetto viene realizzato come insieme di programmi e/o moduli nel linguaggio di programmazione scelto (o nei linguaggi di programmazione scelti)

“The fastest line of code to develop is a line of code you don't have to write” - Jeff Tash

- Gestione delle **versioni** del software



Realizzazione e Collaudo dei Moduli

- Verifica del software

- Analisi statica
- Analisi dinamica
- *Debugging*



- Collaudo dei moduli (**unit testing**)
verifica che un modulo soddisfi le specifiche di progetto

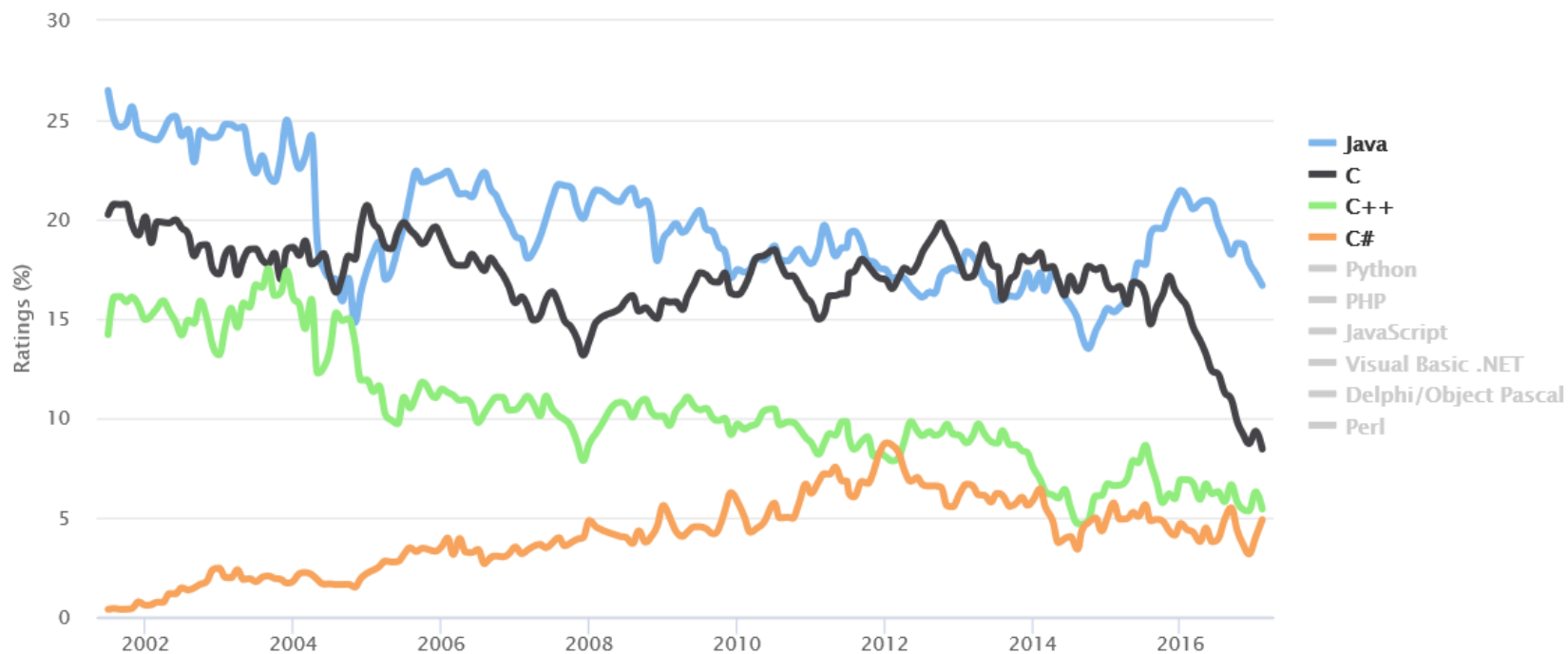


Processo di Sviluppo del Software

Tecnologie e Linguaggi

TIOBE Programming Community Index

Source: www.tiobe.com

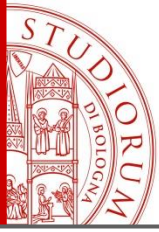




Processo di Sviluppo del Software

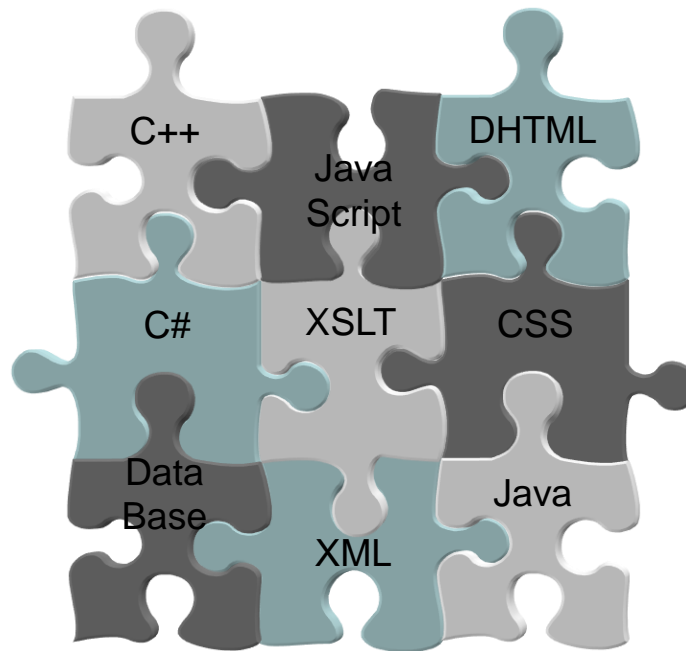
Tecnologie e Linguaggi

- HTML – *Hypertext Markup Language*
- DHTML – *Dynamic HTML*
- CSS – *Cascading Style Sheets*
- Javascript (Typescript), PHP, Python
- XML – *eXtensible Markup Language*
- DTD – *Document Type Definition*
- XSD – *XML Schema Definition*
- XSL – *eXtensible Stylesheet Language*
- XSLT – *XSL for Transformations*
- COM, COM++, DCOM, CORBA, EJB
- .NET, ADO, LinQ, XAML, WPF, WF, WCF
- DBMS, SQL, ORM



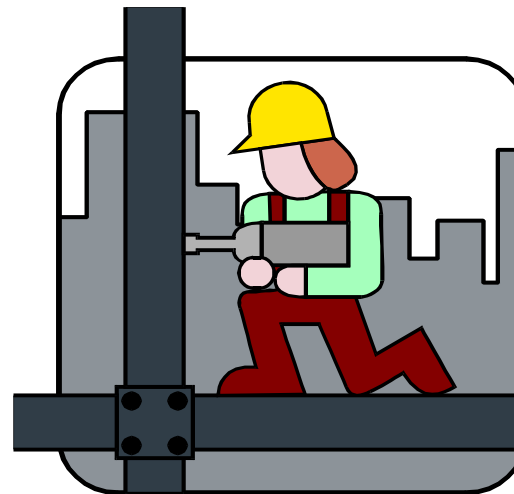
Processo di Sviluppo del Software

Tecnologie e Linguaggi



Integrazione e Collaudo del Sistema

- Si integrano i singoli moduli e/o programmi tra loro e si esegue il test del sistema completo per assicurarsi che le specifiche siano soddisfatte
- **alfa test**
 - il sistema è rilasciato per l'uso, ma all'interno dell'organizzazione del produttore
- **beta test**
 - si ha un rilascio controllato a pochi e selezionati utenti del prodotto





Processo di Sviluppo del Software

Controllo della Qualità

- Uso di programmi di test
- Sollecitazione dei programmi (condizioni limite)
- Controllo dell'utilizzo di standard prestabiliti
- ...



Processo di Sviluppo del Software

Installazione e *Training*

- *Deployment*

Il sistema software viene:

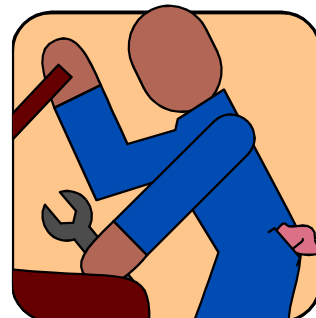
- consegnato al cliente
- installato
- messo in funzione

- *Training*

Processo di Sviluppo del Software

Utilizzo e Manutenzione

- Il sistema viene utilizzato...
- Fase di **manutenzione**
è la fase più lunga del ciclo di vita
di un prodotto software
50-80% dei costi complessivi





Processo di Sviluppo del Software

Utilizzo e Manutenzione

- **Manutenzione correttiva**
correzione degli errori che non sono stati scoperti nelle fasi precedenti
- **Manutenzione adattativa**
aumento dei servizi forniti dal sistema in seguito alla definizione di nuovi requisiti
- **Manutenzione perfettiva**
miglioramento delle caratteristiche delle unità del sistema



Processo di Sviluppo del Software

Utilizzo e Manutenzione

- Spesso il software non viene progettato per essere modificato facilmente
- Vengono apportate modifiche **direttamente sui programmi**, senza modificare
 - la documentazione di progetto
 - la documentazione di test
 - la specifica dei requisiti
 - ...

Processo di Sviluppo del Software

Utilizzo e Manutenzione

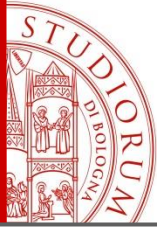
- *Re-ingegnerizzazione* del software
 - Ristrutturazione del codice (*refactoring*)
 - Conversione del linguaggio
 - *Reverse engineering*



Processo di Sviluppo del Software

- Pianificazione, controllo e gestione del processo di sviluppo
 - Analisi dei costi
 - Gestione del gruppo di lavoro
- Gestione dei rischi
 - Rischi relativi ai requisiti
 - Rischi legati alle risorse umane
 - Rischi tecnologici
 - Rischi politici





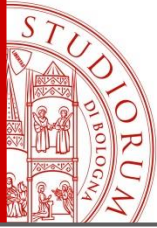
Fattori di Qualità del Software

- Costruire software con caratteristiche di qualità non significa solo produrre codice che “**funziona**” ma anche dare risposta a due domande fondamentali:
 - è stata costruita la cosa giusta? (*validazione*)
 - il prodotto è stato costruito nel modo giusto? (*verifica*)
- Il processo di revisione può avvenire a prodotto costruito o in itinere e secondo approcci a **black box** o a **white box**



Fattori di Qualità del Software

- **Qualità esterne** percepibili da un osservatore esterno che esamina il prodotto software come se fosse una scatola nera (*black-box*)
 - Affidabilità
 - Facilità d'uso
 - Velocità
 - ...
- Devono essere garantite



Fattori di Qualità del Software

- **Qualità interne**
osservabili esaminando la struttura interna del prodotto software, come se questo fosse una scatola trasparente (*white-box*)
 - Modularità
 - Leggibilità
 - ...
- Influenzano le qualità esterne
- Sono un modo per realizzare le qualità esterne

Fattori di Qualità del Software

Correttezza (*Correctness*)

- Data una definizione dei requisiti che il software deve soddisfare, il software si dice corretto se rispetta tali requisiti



I requisiti specificati risultano spesso incompleti rispetto ai requisiti reali



Fattori di Qualità del Software

Robustezza (*Robustness*)

- Il software si dice robusto se si *comporta in maniera accettabile* anche in corrispondenza di situazioni anomale e comunque non specificate nei requisiti
- Nel caso di situazioni anomale, il software *non deve causare disastri* (perdita di dati o peggio)
 - o termina l'esecuzione in modo pulito
 - o entra in una modalità particolare, in cui non sono più attive alcune funzionalità (*graceful degradation mode*)

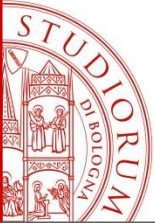


Fattori di Qualità del Software

Affidabilità (*Reliability*)

- Il software si dice affidabile se
 - le funzionalità offerte corrispondono ai requisiti (il software è corretto)
 - in caso di guasto, non produce danni fisici o economici (il software è robusto)

Affidabilità = Correttezza + Robustezza



Fattori di Qualità del Software

Facilità d'Uso (*Ease of Use*)

- Facilità con cui l'utilizzatore del software è in grado di:
 - Imparare ad usare il sistema
 - Utilizzare il sistema
 - Fornire i dati da elaborare
 - Interpretare i risultati
 - Gestire condizioni di errore



Fattori di Qualità del Software

Efficienza (*Efficiency*)

- Buon utilizzo delle risorse hardware:
 - Processori (tempo di calcolo)
 - Memoria principale (occupazione di memoria)
 - Memorie secondarie
 - Canali di comunicazione



Fattori di Qualità del Software

Estensibilità (*Extensibility*)

- Facilità con cui il software **può essere modificato** per soddisfare nuovi requisiti
- La modifica di programmi di piccole dimensioni non è un problema serio
- I sistemi software di dimensioni medie e grandi possono soffrire di fragilità strutturale:
 - modificando un singolo elemento della struttura si rischia di far collassare l'intera struttura



Fattori di Qualità del Software

Estensibilità (*Extensibility*)

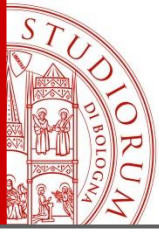
- Due principi essenziali:
 - Semplicità architeturale
 - più l'architettura del sistema è semplice
 - più è facile da modificare
 - Modularità e decentralizzazione
 - più il sistema è suddiviso in moduli autonomi
 - più è facile che una modifica coinvolga un numero limitato di moduli



Fattori di Qualità del Software

Riusabilità (*Reusability*)

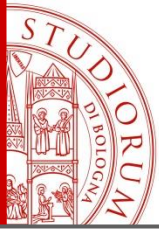
- Il software è riusabile se può essere **riutilizzato completamente o in parte** in nuove applicazioni
- Permette di non dover reinventare soluzioni a problemi già affrontati e risolti
- Influenza tutte le altre caratteristiche dei prodotti software



Fattori di Qualità del Software

Verificabilità (*Verifiability*)

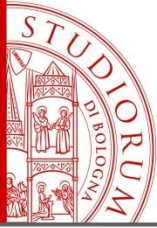
- Facilità con cui il prodotto software può essere sottoposto a test



Fattori di Qualità del Software

Portabilità (*Portability*)

- Facilità con cui il prodotto software può essere trasferito su altre architetture hardware e/o software



Fattori di Qualità del Software

- Molte caratteristiche sono definibili soltanto in **maniera intuitiva e poco rigorosa**
- Alcune caratteristiche sono **in contrapposizione** ad esempio: efficienza vs portabilità
- L'**importanza** dell'una o dell'altra caratteristica varia (può variare) a seconda del settore applicativo
- Il **costo del software** aumenta enormemente se è richiesto un livello molto alto di una qualunque di tali caratteristiche